

Ref #	Hits	Search Query	DBs	Default Operator	Plurals	Time Stamp
L2	12	719/328-310.ccls. and (interfac\$3 or communicat\$3 or implement\$3) with (heterogeneous or different or various or assorted or variety) with (api or (application adj programming adj interface)) with (map\$4 or translat\$3 or transform\$3 or interpret\$3)	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2005/07/12 08:59
L4	32	719/328-310.ccls. and (api or (application adj programming adj interface)) near framework	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2005/07/12 08:59
S7	30	719/328.ccls. and (universal or public or non?proprietary or generic or standard\$3) adj (api or (application adj programming adj interface))	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2005/07/11 12:13
S8	24	719/328.ccls. and (universal or public or non?proprietary or generic or standard\$3) adj (api or (application adj programming adj interface)) and object	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2004/12/16 08:18
S9	12	719/328.ccls. and (universal or public or non?proprietary or generic or standard\$3) adj (api or (application adj programming adj interface)) and (map\$4 with object)	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2004/12/15 16:15
S14	10	work\$1flow with (interface or api) same (object with (map\$4 or adapter))	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2004/12/16 16:54
S15	17	work\$1flow with (interface or api) same (object same (map\$4 or adapter))	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2004/12/15 16:51
S17	31	((application adj2 interface) or api) same ((object same (map\$4 or adapter)) with (native or prioprietary or legacy))	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2004/12/16 09:07
S21	101	((interface) or api) same ((object same (map\$4 or adapter)) with (native or prioprietary or legacy))	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2004/12/16 08:35

S23	65	((interface) or api) same ((object same (map\$4 or adapter)) near3 (native or proprietary or legacy or public or standard\$6 or generic))	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2004/12/16 08:37
S25	43	((interface) or api) same ((object same (map\$4 or adapter)) near2 (native or proprietary or legacy or custom or public or standard\$6 or generic))	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2004/12/16 08:43
S27	33	work\$1flow adj3 (interface or api) and ((native or proprietary or legacy or custom or local) with object)	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2004/12/16 09:36
S28	8	work\$1flow adj3 (interface or api) with (native or proprietary or legacy or custom or local)	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2004/12/16 11:58
S29	49	(work\$1flow or wfmc) with (interface or api) with (native or proprietary or legacy or custom or local)	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2004/12/16 14:39
S30	51	(work\$1flow or wfmc) with (interface or api) with (native or proprietary or legacy or custom or local or heterogen\$4)	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2004/12/16 11:59
S32	38	(work\$1flow or wfmc) with (interface or api) with (native or proprietary or legacy or custom or local or heterogen\$4) and ((@ad<"20011129") or (@prad<"20011129") or (@rlad<"20011129"))	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2004/12/16 12:33
S34	12	((work\$1flow or wfmc) near heterogen\$4) and (interface or api) and ((@ad<"20011129") or (@prad<"20011129") or (@rlad<"20011129"))	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2004/12/16 12:37
S35	128	((work\$1flow or wfmc) near (heterogen\$4 or vari\$4 or different or vendor\$1speci\$4)) and (interface or api) and ((@ad<"20011129") or (@prad<"20011129") or (@rlad<"20011129"))	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2004/12/16 12:44

S36	9	((work\$1flow or wfmc) near (heterogen\$4 or vari\$4 or different or vendor\$1speci\$3)) same (interface or api) and ((@ad<"20011129") or (@prad<"20011129") or (@rlad<"20011129"))	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2004/12/16 12:44
S37	4	S36 not S34	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2004/12/16 12:44
S39	40	work\$1flow with (interface or api) with (map\$4 or adapter)	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2004/12/16 17:04
S40	5	work\$1flow with (interface or api) with (map\$4 or adapter or translat\$4) near3 object	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2004/12/16 17:04
S43	1	"20030004770"	US-PGPUB	OR	OFF	2004/12/21 09:18
S44	1	"20030004771"	US-PGPUB	OR	OFF	2004/12/21 09:18
S50	129	object with (process adj instance)	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2004/12/21 14:19
S51	2	(object with content) with (process adj instance)	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2004/12/21 14:28
S53	8	(work\$1flow adj management) same (content adj management)	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2004/12/21 15:10
S56	90	work\$1flow with (content adj management)	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2004/12/21 15:25
S57	1	"20020133635"	US-PGPUB; USPAT	OR	OFF	2005/07/11 09:54
S58	1	"20020129016"	US-PGPUB; USPAT	OR	OFF	2005/07/11 09:54

S59	3	719/328.ccls. and (universal or public or non?proprietary or generic or standard\$3) near2 (api or (application adj programming adj interface)) with native	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2005/07/11 12:15
S60	7	719/328.ccls. and (universal or public or non?proprietary or generic or standard\$3 or common) near2 (api or (application adj programming adj interface) or interface) with native	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2005/07/11 12:27
S61	2	"6,854,120".pn.	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2005/07/11 12:57
S62	14	(map\$4 or translat\$3) with generic near3 native	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2005/07/11 13:18
S63	53	(map\$4 or translat\$3 or transform\$3 or interpret\$3) with generic with native	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2005/07/11 15:52
S64	4	719/328.ccls. and (map\$4 or translat\$3 or transform\$3 or interpret\$3) with generic with native	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2005/07/11 15:52
S66	5	(719/328-310).ccls. and (map\$4 or translat\$3 or transform\$3 or interpret\$3) with generic with native	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2005/07/11 14:41
S67	1	S66 not S64	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2005/07/11 13:57
S68	34	719/328.ccls. and (heterogeneous or different) adj (api or (application adj programming adj interface))	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2005/07/11 14:35
S69	27	719/328-310.ccls. and (heterogeneous or different) with (api or (application adj programming adj interface)) with (map\$4 or translat\$3 or transform\$3 or interpret\$3)	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2005/07/12 08:38

S70	16	719/328-310.ccls. and (heterogeneous or different) with (api or (application adj programming adj interface)) with (generic or public)	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2005/07/11 15:08
S71	31	719/328-310.ccls. and (heterogeneous or different) with (api or (application adj programming adj interface) or interface) with (generic or public)	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2005/07/11 15:10
S72	15	(map\$4 or translat\$3 or transform\$3 or interpret\$3) with generic with native with java	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2005/07/11 15:53

**PORTAL**  
USPTO

[Subscribe \(Full Service\)](#) [Register \(Limited Service, Free\)](#) [Login](#)

Search:  The ACM Digital Library  The Guide

generic application programming interface api

THE ACM DIGITAL LIBRARY

Feedback Report a problem Satisfaction survey

Terms used **generic application programming interface api**

Found 82,941 of 157,873

Sort results  relevance  Save results to a BinderDisplay results  expanded form  Search Tips Open results in a new windowTry an [Advanced Search](#)Try this search in [The ACM Guide](#)

Results 1 - 20 of 200

Result page: **1** [2](#) [3](#) [4](#) [5](#) [6](#) [7](#) [8](#) [9](#) [10](#) [next](#)

Best 200 shown

Relevance scale 

**1** [Next-generation generic programming and its application to sparse matrix computations](#) 

Nikolay Mateev, Keshav Pingali, Paul Stodghill, Vladimir Kotlyar

May 2000 **Proceedings of the 14th international conference on Supercomputing**Full text available:  [pdf\(1.06 MB\)](#)Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#)

The contributions of this paper are the following. We introduce a new variety of generic programming in which algorithm implementors use a different API than data structure designers, the gap between the API's being bridged by restructuring compilers. One view of this approach is that it exploits restructuring compiler technology to perform a novel kind of template instantiation. We demonstrate the usefulness of this new generic programming technology ...

**2** [An object oriented application/programmer interface for network programming](#) 

Jennifer Howell, Ming Shu, Robert Wohlfarth

March 1993 **Proceedings of the 1993 ACM/SIGAPP symposium on Applied computing: states of the art and practice**Full text available:  [pdf\(685.85 KB\)](#) Additional Information: [full citation](#), [references](#), [index terms](#)

**3** [Development of an OSI application layer protocol interface](#) 

K. Fong, J. Reinstedler

July 1989 **ACM SIGCOMM Computer Communication Review**, Volume 19 Issue 3Full text available:  [pdf\(1.93 MB\)](#) Additional Information: [full citation](#), [index terms](#)

**4** [Using metalevel techniques in a flexible toolkit for CSCW applications](#) 

Paul Dourish

June 1998 **ACM Transactions on Computer-Human Interaction (TOCHI)**, Volume 5 Issue 2Full text available:  [pdf\(292.97 KB\)](#) Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#)

Ideally, software toolkits for collaborative applications should provide generic, reusable components, applicable in a wide range of circumstances, which software developers can assemble to produce new applications. However, the nature of CSCW applications and the mechanics of group interaction present a problem. Group interactions are significantly constrained by the structure of the underlying infrastructure, below the level at which toolkits typically offer control. This article describe ...

**Keywords:** consistency control, consistency guarantees, data distribution, divergency, metalevel programming, open implementation, software architecture

5 **Technical papers:** NETKIT: a software component-based approach to programmable networking 

Geoff Coulson, Gordon Blair, David Hutchison, Ackbar Joolia, Kevin Lee, Jo Ueyama, Antonio Gomes, Yimin Ye

October 2003 **ACM SIGCOMM Computer Communication Review**, Volume 33 Issue 5

Full text available:  pdf(316.64 KB) Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#)

While there has already been significant research in support of openness and programmability in networks, this paper argues that there remains a need for generic support for the integrated development, deployment and management of programmable networking software. We further argue that this support should explicitly address the management of run-time reconfiguration of systems, and should be independent of any particular programming paradigm (e.g. active networking or open signaling), programmin ...

**Keywords:** components, middleware, programmable networking, reflection

6 **Concrete syntax for objects: domain-specific language embedding and assimilation without restrictions** 

Martin Bravenboer, Eelco Visser

October 2004 **ACM SIGPLAN Notices , Proceedings of the 19th annual ACM SIGPLAN Conference on Object-oriented programming, systems, languages, and applications**, Volume 39 Issue 10

Full text available:  pdf(379.91 KB) Additional Information: [full citation](#), [abstract](#), [references](#), [index terms](#)

Application programmer's interfaces give access to domain knowledge encapsulated in class libraries without providing the appropriate notation for expressing domain composition. Since object-oriented languages are designed for extensibility and reuse, the language constructs are often sufficient for expressing domain abstractions at the semantic level. However, they do not provide the right abstractions at the syntactic level. In this paper we describe MetaBorg, a method for providing <i> ...

**Keywords:** MetaBorg, SDF, concrete object syntax, domain-specific languages, embedded languages, extensible syntax, meta programming, rewriting, stratego, syntax extension

7 **Mirrors: design principles for meta-level facilities of object-oriented programming languages** 

Gilad Bracha, David Ungar

October 2004 **ACM SIGPLAN Notices , Proceedings of the 19th annual ACM SIGPLAN Conference on Object-oriented programming, systems, languages, and applications**, Volume 39 Issue 10

Full text available:  pdf(303.44 KB) Additional Information: [full citation](#), [abstract](#), [references](#), [index terms](#)

We identify three design principles for reflection and metaprogramming facilities in object oriented programming languages. <i>Encapsulation</i>: meta-level facilities must encapsulate their implementation. <i>Stratification</i>: meta-level facilities must be separated from base-level functionality. <i>Ontological correspondence</i>: the ontology of meta-level facilities should correspond to the ontology of the language they manipulate. Traditional/mainstream reflectiv ...

**Keywords:** java, metaprogramming, mirrors, reflection, self, smalltalk

8 DROOPI: towards a generic middleware

Thomas Quinot, Fabrice Kordon, Laurent Pautet  
 June 2001 **ACM SIGAda Ada Letters**, Volume XXI Issue 2

Full text available:  pdf(1.34 MB) Additional Information: [full citation](#), [abstract](#), [references](#)

This paper presents our work to bridge the Ada 95 Distributed Systems Annex (DSA) and CORBA to take advantages of both environments facilities. Our project consists in two successive steps. The first one is CIAO, a DSA to CORBA translator. The second one aims at the definition of a generic middleware to be customized to DSA and CORBA. We propose a definition and an architecture of services for a generic middleware, *DROOPI*, and explain how it can be customized according various cr ...

9 Applications of JAVA programming language to database management

Bradley F. Burton, Victor W. Marek  
 March 1998 **ACM SIGMOD Record**, Volume 27 Issue 1

Full text available:  pdf(258.27 KB) Additional Information: [full citation](#), [index terms](#)

10 Cg: a system for programming graphics hardware in a C-like language

William R. Mark, R. Steven Glanville, Kurt Akeley, Mark J. Kilgard  
 July 2003 **ACM Transactions on Graphics (TOG)**, Volume 22 Issue 3

Full text available:  pdf(2.57 MB) Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#)

The latest real-time graphics architectures include programmable floating-point vertex and fragment processors, with support for data-dependent control flow in the vertex processor. We present a programming language and a supporting system that are designed for programming these stream processors. The language follows the philosophy of C, in that it is a hardware-oriented, general-purpose language, rather than an application-specific shading language. The language includes a variety of facilitit ...

11 Systems: Leveraging single-user applications for multi-user collaboration: the coword approach

Steven Xia, David Sun, Chengzheng Sun, David Chen, Haifeng Shen  
 November 2004 **Proceedings of the 2004 ACM conference on Computer supported cooperative work**

Full text available:  pdf(536.98 KB) Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#)

Single-user interactive computer applications are pervasive in our daily lives and work. Leveraging single-user applications for multi-user collaboration has the potential to significantly increase the availability and improve the usability of collaborative applications. In this paper, we report an innovative <i>transparent adaptation</i> approach for this purpose. The basic idea is to adapt the single-user application programming interface to the data and operational models of the un ...

**Keywords:** application sharing, operational transformation, real-time collaborative word processor, transparent adaptation

12 Distributed interface bits: dynamic dialogue composition from ambient computing resources

Anthony Savidis, Constantine Stephanidis  
 May 2005 **Personal and Ubiquitous Computing**, Volume 9 Issue 3

Full text available:  pdf(1.70 MB) Additional Information: [full citation](#), [abstract](#)

This paper discusses a particular issue in the context of disappearing computing, namely, user mobility. Mobile users may carry with them a variety of wireless gadgets while being immersed in a physical environment encompassing numerous computing devices. In such a

situation, it is most likely that the number and type of devices may dynamically vary during interactions. The Voyager development framework supports the implementation of ambient dialogues, i.e., dynamically distributed user Interfac ...

**Keywords:** Abstract dialogue elements, Adaptive interaction, Ambient dialogues, Dynamic user interface composition, Wearable interfaces

**13** Hypermedia systems: Structure and behavior awareness in themis 

Kenneth M. Anderson, Susanne A. Sherba, William V. Leptihien

August 2003 **Proceedings of the fourteenth ACM conference on Hypertext and hypermedia**

Full text available:  [pdf\(291.23 KB\)](#)

Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#)

Structural computing provides techniques and tools to ease the task of developing application infrastructure; infrastructure that provides common services such as persistence, naming, distribution, navigational hypermedia, etc., over a set of application-specific or domain-specific *structures*. Within structural computing, "structure" refers to a combination of data together with relationships pertaining to that data. Structure servers support the specification and manipulation of structur ...

**Keywords:** awareness, behavior, structural computing, structure, themis

**14** Special session: Design and programming of embedded multiprocessors: an interface-centric approach 

Pieter van der Wolf, Erwin de Kock, Tomas Henriksson, Wido Kruijzer, Gerben Essink

September 2004 **Proceedings of the 2nd IEEE/ACM/IFIP international conference on Hardware/software codesign and system synthesis**

Full text available:  [pdf\(377.96 KB\)](#)

Additional Information: [full citation](#), [abstract](#), [references](#), [index terms](#)

We present design technology for the structured design and programming of embedded multi-processor systems. It comprises a task-level interface that can be used both for developing parallel application models and as a platform interface for implementing applications on multi-processor architectures. Associated mapping technology supports refinement of application models towards implementation. By linking application development and implementation aspects, the technology integrates the specificat ...

**Keywords:** code transformation, media processing, multiprocessor mapping, platform interface, system design method, task-level interface

**15** Pluggable verification modules: an extensible protection mechanism for the JVM 

Philip W. L. Fong

October 2004 **ACM SIGPLAN Notices , Proceedings of the 19th annual ACM SIGPLAN Conference on Object-oriented programming, systems, languages, and applications**, Volume 39 Issue 10

Full text available:  [pdf\(224.39 KB\)](#)

Additional Information: [full citation](#), [abstract](#), [references](#), [index terms](#)

Through the design and implementation of a JVM that supports Pluggable Verification Modules (PVMs), the idea of an extensible protection mechanism is entertained. Link-time bytecode verification becomes a pluggable service that can be readily replaced, reconfigured and augmented. Application-specific verification services can be safely introduced into the dynamic linking process of the JVM. This feature is enabled by the adoption of a previously proposed modular verification architecture, Pro ...

**Keywords:** Aegis VM, Java virtual machine, bytecode verification, extensible protection mechanism, extensible systems, mobile code security, pluggable verification modules, proof linking

**16 PRIME—toward process-integrated modeling environments: 1**

Klaus Pohl, Klaus Weidenhaupt, Ralf Dömges, Peter Haumer, Matthias Jarke, Ralf Klamma  
 October 1999 **ACM Transactions on Software Engineering and Methodology (TOSEM)**,  
 Volume 8 Issue 4

Full text available:  [pdf\(1.15 MB\)](#)Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#), [review](#)

Research in process-centered environments (PCEs) has focused on project management support and has neglected method guidance for the engineers performing the (software) engineering process. It has been dominated by the search for suitable process-modeling languages and enactment mechanisms. The consequences of process orientation on the computer-based engineering environments, i.e., the interactive tools used during process performance, have been studied much less. In this article, we prese ...

**Keywords:** PRIME, method guidance, process modeling, process-centered environments, process-integrated environments, process-sensitive tools, tool integration, tool modeling

**17 Fast detection of communication patterns in distributed executions**

Thomas Kunz, Michiel F. H. Seuren  
 November 1997 **Proceedings of the 1997 conference of the Centre for Advanced Studies on Collaborative research**

Full text available:  [pdf\(4.21 MB\)](#) Additional Information: [full citation](#), [abstract](#), [references](#), [index terms](#)

Understanding distributed applications is a tedious and difficult task. Visualizations based on process-time diagrams are often used to obtain a better understanding of the execution of the application. The visualization tool we use is Poet, an event tracer developed at the University of Waterloo. However, these diagrams are often very complex and do not provide the user with the desired overview of the application. In our experience, such tools display repeated occurrences of non-trivial commun ...

**18 Technical papers: component technologies: Pluggable reflection: decoupling meta-interface and implementation**

David H. Lorenz, John Vlissides  
 May 2003 **Proceedings of the 25th International Conference on Software Engineering**

Full text available:   [pdf\(1.10 MB\)](#) [DOI](#) Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#)  
[Publisher Site](#)

Reflection remains a second-class citizen in current programming models, where it's assumed to be imperative and tightly bound to its implementation. In contrast, most object-oriented APIs allow interfaces to vary independently of their implementations. Components take this separation a step further by describing unforeseeable attributes---the key to pluggable third-party components. This paper describes how reflection can benefit from a similar evolutionary path.

**19 System support for pervasive applications**

Robert Grimm, Janet Davis, Eric Lemar, Adam Macbeth, Steven Swanson, Thomas Anderson, Brian Bershad, Gaetano Borriello, Steven Gribble, David Wetherall  
 November 2004 **ACM Transactions on Computer Systems (TOCS)**, Volume 22 Issue 4

Full text available:  [pdf\(1.82 MB\)](#) Additional Information: [full citation](#), [abstract](#), [references](#), [index terms](#)

Pervasive computing provides an attractive vision for the future of computing. Computational power will be available everywhere. Mobile and stationary devices will dynamically connect and coordinate to seamlessly help people in accomplishing their tasks. For this vision to become a reality, developers must build applications that constantly adapt to a highly dynamic computing environment. To make the developers' task feasible, we present a system architecture for pervasive computing, called & ...

**Keywords:** Asynchronous events, checkpointing, discovery, logic/operation pattern,

migration, one.world, pervasive computing, structured I/O, tuples, ubiquitous computing

## 20 Non-invasive adaptation of black-box user interfaces



D. Rose, S. Stegmaier, G. Reina, D. Weiskopf, T. Ertl

February 2003 **Proceedings of the Fourth Australian user interface conference on User interfaces 2003 - Volume 18 CRPITS '03**

Full text available: [pdf\(3.62 MB\)](#) Additional Information: [full citation](#), [abstract](#), [references](#), [index terms](#)

In this paper a new method for the non-invasive adaptation of user interfaces is presented. The main idea is not to implement the user interface toolkit as an API, but instead as an object file that redefines the functionality of the API of an already existing toolkit in a generic way based on a so-called preloading technique. Compared to common approaches, the presented method allows us to evaluate prototypical user interfaces with a large number of real-world applications with very little effo ...

**Keywords:** UI evaluation, UI prototyping, UI retargeting, menu navigation, user interfaces

Results 1 - 20 of 200

Result page: [1](#) [2](#) [3](#) [4](#) [5](#) [6](#) [7](#) [8](#) [9](#) [10](#) [next](#)

The ACM Portal is published by the Association for Computing Machinery. Copyright © 2005 ACM, Inc.

[Terms of Usage](#) [Privacy Policy](#) [Code of Ethics](#) [Contact Us](#)

Useful downloads: [Adobe Acrobat](#) [QuickTime](#) [Windows Media Player](#) [Real Player](#)